REMARKS

No claims have been amended, added, or cancelled. Therefore, claims 1-48 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(b) Rejection:

The Office Action rejected claims 1-48 under 35 U.S.C. § 102(b) as being anticipated by Hinks et al. (U.S. Patent 5,678,039) (hereinafter "Hinks"). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, Hinks does not disclose identifying at least one token within the document and identifying a localizable string within the token and further fails to disclose creating a second file including non-localizable data from the document; and merging the first file and the second file. Instead, Hinks teaches a software translation kit for translating application programs into different languages. Hinks' system involves extracting translatable resources (i.e. strings, dialog boxes, menu items, etc) from a program, providing a user interface to allow a user to translate those resources, and rebuilding the original program. Furthermore, Hinks teaches that the Export/Import module of his system includes a parsing engine "to extract strings and translatable information from application programs", not within a markup-language document (Hinks, column 3, lines 1-3).

Contrary to the Examiner's assertion, Hinks does not disclose, identifying at least one token within the markup language document and identifying a localizable string within the token. The Examiner cites column 7, lines 8-10 and column 8, lines 10-12 of Hinks. However, The cited passage does not describe identifying a token within a markup language document and identifying a localizable string within the token. Instead, the first cited passage (column 7, lines 8-10) describes how some translation systems use a database that includes a token "that uniquely identifies a string to be translated." The

second cited passage (column 8, lines 10-12) refers to how a user translates strings using various resource editors (string editor, menu editor, dialog editor, etc) and how the translations are stored in Hinks' translation table. Thus, Hinks does not describe identifying a token within a markup language document and identifying a localizable string *within the token*. Instead, Hinks describes tokens that are unique identifiers for translatable strings but which do not include localizable strings. For instance, Hinks states that his tokens are collections of fields that make a given record in the translation table unique so that the record can be used in data manipulation, when merging back to the target program's sources, or when upgrading between versions (Hinks, column 17, lines 5-9). Nowhere does Hinks describe identifying a localizable string within a token. As noted above, Hinks' tokens are related to individual records in his translation table database and do not include translatable or localizable strings. Thus, Hinks fails to teach identifying a token within a markup language document and identifying a localizable string *within the token*.

Additionally, Hinks fails to disclose creating a second file including non-localizable data from the markup language document; and merging the first file and the second file. The Examiner cites column 17, lines 4-12 and lines 9-12 of Hinks. However, the cited portion of Hinks does not mention creating a second file including non-localizable data. Instead, Hinks describes the content of his translation table, "which comprises ... three major field types: token, English/source, and translation" (Hinks, column 17, lines 2-4). Hinks also describes how the token field is a collection of one or more fields that make the given record unique, how the English/source field stores translatable strings, dialog coordinates, accelerators, etc. that can be translated, and how the translation fields are fields where the translator (user) enters the translated strings, coordinates, etc. The Examiner's cited passage does not mention a second file including non-localizable data. Instead, Hinks teaches that once the end-user translator has completed the task of translating the resources, the translated text is merged back to [the] sources" (Hinks, column 8, lines 14-16). Specifically, Hinks teaches that a translated resource file is created and subsequently, "the target product is rebuilt with the new sources" (Hinks, column 8, lines 24-28). Hinks describes two methods for rebuilding the

target program to include the translated data. In one approach, the translated resource file is stored with the other program sources and the program is re-compiled. In the other approach, a resource compiler is used to compile the translated resource file and to bind the compiled resources back into the target program. Thus, nowhere does Hinks disclose creating a second file including non-localizable data from the document and merging the first file and the second file. Instead, as noted above, Hinks teaches extracting the translatable data from a program, translating those data, and re-compiling the target program to include the translated data.

For at least the reasons presented above, the rejection of claim 1 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 1 also apply to claims 10 and 19.

Regarding claim 28, Hinks fails to disclose identifying at least one token within the markup language document and identifying a localizable string within the token. As with the rejection of claim 1, discussed above, the Examiner cites column 7, lines 8-10 and column 8, lines 10-12 of Hinks. However, as noted above, the cited passages do not describe identifying a token with a markup language document and identifying a localizable string within the token. Instead, the cited passages describe how some translation systems use a database that includes a token "that uniquely identifies a string to be translated" and how a user translates strings using various resource editors (string editor, menu editor, dialog editor, etc) and how the translations are stored in Hinks' translation table. For a more detailed discussion regarding Hinks' failure to describe identifying at least one token within the markup language document and identifying a localizable string within the token, please refer to the discussion of claim 1 above.

In further regard to claim 28, Hinks additionally fails to disclose extracting non-localizable data from the markup language document and merging the extracted non-localizable data with at least one of the translated extracted localizable strings and the extracted localizable string. The Examiner cites column 17, lines 4-5 and lines 9-12 and argues that Hinks teaches "an export module able to place the English/source non-

translatable information into a translation table." However, at the Examiner's cited passage, Hinks specifically states that the English/source field of the translation table stores, "entries [that] represent the elements that will differ between the 'source language' base product and the 'translated' product" (Hinks, column 17, lines 9-12). Thus, in direct contrast to the Examiner's assertion, the English/source field cited by the Examiner specifically stores the translatable portions of the source program.

The Examiner also cites column 7, lines 17-20 as teaching merging extracted non-localizable data with translated localizable strings. However, the cited passage describes how Hinks' system is data-centric in that "[d]uring the translation process the focus is on the data: what data need to be extracted from resource files, translated into the local language, and reinserted into the sources" (Hinks, column 7, lines 17-20). Thus, the Examiner cites passage does not mention anything about merging *extracted* non-localizable data with translated localizable strings. Instead, the cited passage refers only to extracting translatable data, translating that data, and *reinserting* the translated data back into the target program. Nowhere does Hinks describe merging extracted non-localizable data with translated localizable strings. In fact, the Examiner's cited passage actually supports Applicants' argument that Hinks does not teach extracting non-localizable data from a document and merging that extracted non-localizable data with translated localizable strings.

For at least the reasons presented above, the rejection of claim 28 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 28 also apply to claims 35 and 42.

Additionally, regarding claims 6, 15, 24, 31, 38 and 45, the Examiner fails to provide a proper § 102(b) rejection of these claims. While the Examiner states that claims 1-48 are rejected under § 102(b), the Examiner fails to provide any detailed argument or to cite any portion of Hinks regarding these claims. Additionally, the Examiner rejects claims 6, 15, 24, 31, 38 and 45 under § 103(a) relying upon an obviousness argument in the rejection, thus implying that Hinks fails to disclose the

limitations of these claims. Furthermore, each of claims 6, 15, 24, 31, 38 and 45, is patentable for at least the reasons presented above regarding their respective independent claims. Thus, the § 102(b) of claims 6, 15, 24, 31, 38 and 45 is not supported by the prior art and removal thereof is respectfully requested.

Section 103(a) Rejection:

The Office Action rejected claims 6, 15, 24, 31, 38 and 45 under 35 U.S.C. § 103(a) as being unpatentable over Hinks as applied to claims 1, 10, 19, 28, 35 and 42 above. Applicants respectfully traverse this rejection and submit that claims 6, 15, 24, 31, 38 and 45 are patentable for at least the reasons presented above regarding their respective independent claims.

In regard to the rejections under both § 102(b) and § 103(a), Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-90300/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: June 14, 2005